Trí tuệ nhân tạo trong điều khiển

Convolution Neural Networks
Mạng nơron tích chập

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP

BACHKHOACNCP.COM

Review

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Materials

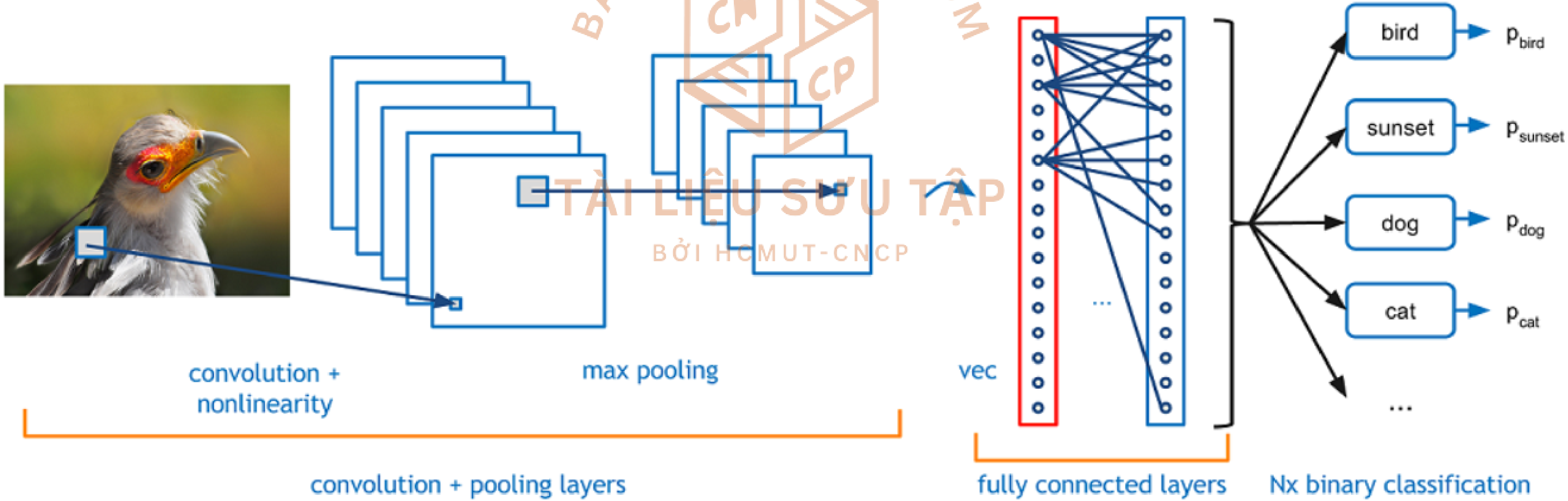
BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP

BACHKHOACNCP.COM

What is CNNs?

✓Convolution Neural Networks

- ❖ CNNs
- ❖ ConvNets



BACHKHOACNCP.COM

Applications

- ✓ Computer vision
  - ❖ Face recognition
  - ❖ Scene labeling
  - ❖ Image classification
  - ❖ Action recognition
  - ❖ Human pose estimation
  - ❖ Document analysis
- ✓ Natural language processing
  - ❖ Speech recognition
  - ❖ Text classification

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Demo

- ✓ Demo:
  - ❖ Image classification: http://cs231n.stanford.edu/
  - ❖ Lane detection
  - ❖ Off-Road Scenes Segmentation and Classification
  - ❖ Self Driving Car
  - ❖ Handwritten digits recognition
  - ❖ Facial expression recognition

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

CNNs vs. Regular NNs

✓ Regular NNs: don't scale well to full images

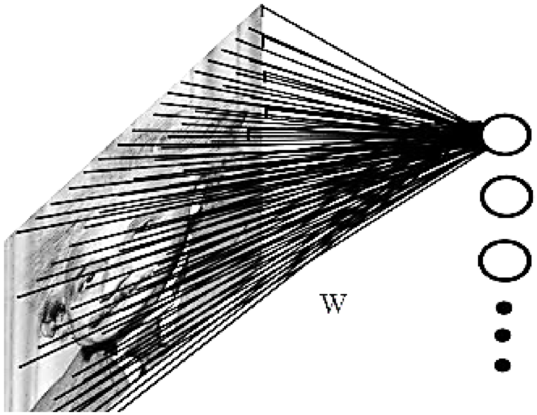
❖ CIFAR-10: image size 32x32x3

➡ A single fully-connected neuron in a first hidden layer has 32*32*3 = 3072 weights

❖ Image size: 200x200x3

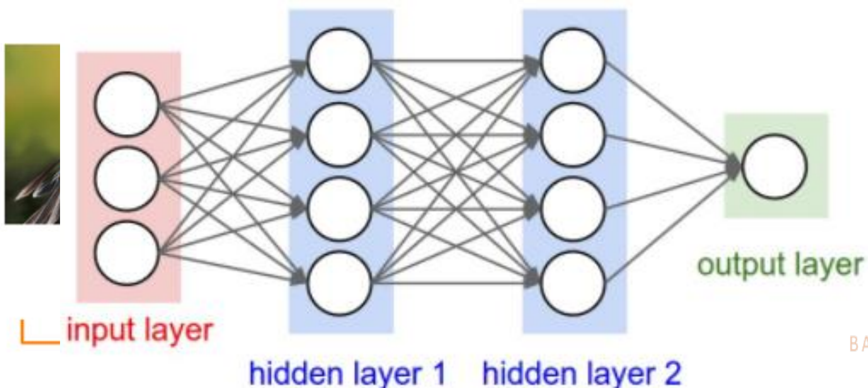
➡ 200*200*3 = 120,000 weights

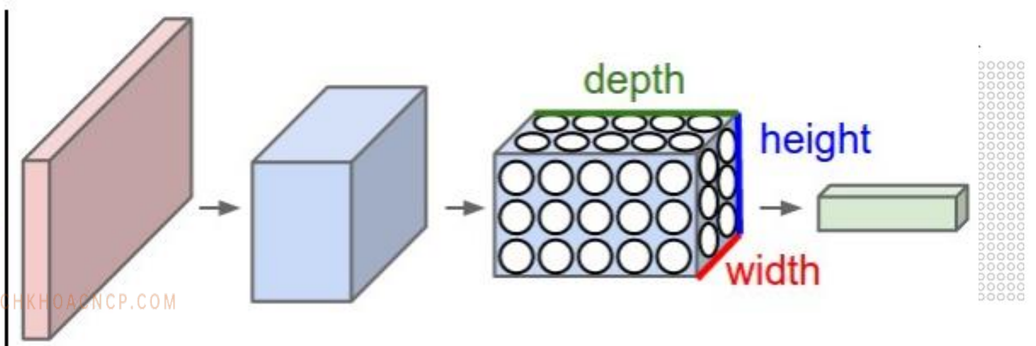
➡ Over fitting



✓ CNNs: Local connectivity

Shared weights





TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

What computers see pictures?

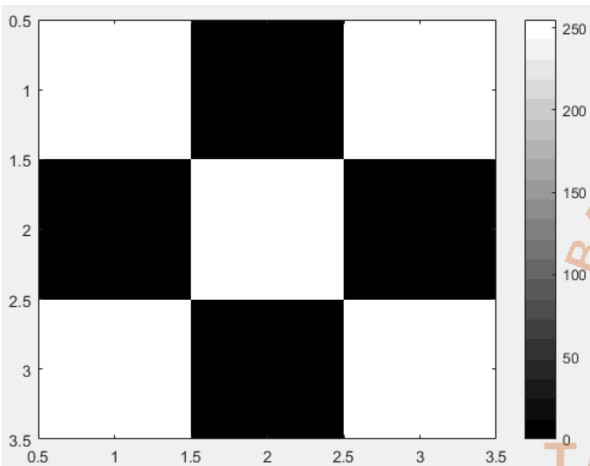
What We See

What Computers See

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

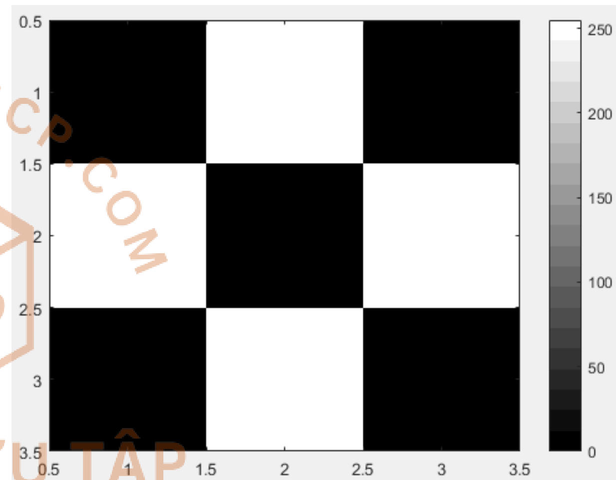
A toy example (forward)

X



255	0	255
0	255	0
255	0	255

O



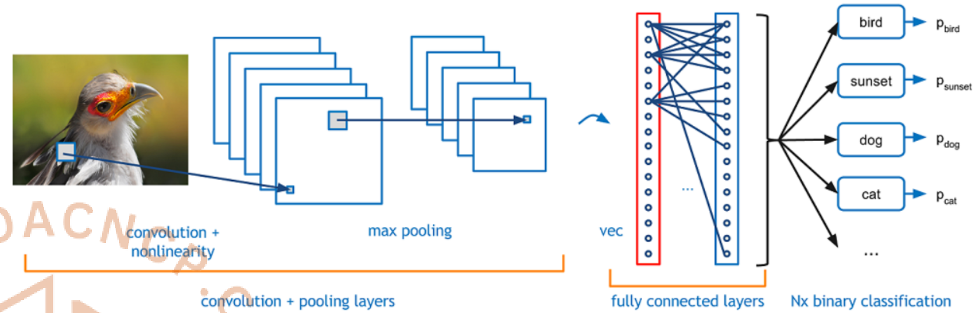
0	255	0
255	0	255
0	255	0

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

A toy example (forward)

255	0	255
0	255	0
255	0	255

0	255	0
255	0	255
0	255	0



Input I

I_{11}	I_{12}	I_{13}
I_{21}	I_{22}	I_{23}
I_{31}	I_{32}	I_{33}

$P(I = X) = ?$

$P(I = O) = ?$

Convolution Layer: 2 kernels (4 weights each) + 2 biases

Fully Connected Layer: 16 weights (regular NN) + 2 biases

k^1_{11}	k^1_{12}
k^1_{21}	k^1_{22}

Kernel 1

k^2_{11}	k^2_{12}
k^2_{21}	k^2_{22}

Kernel 2

28 parameters

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

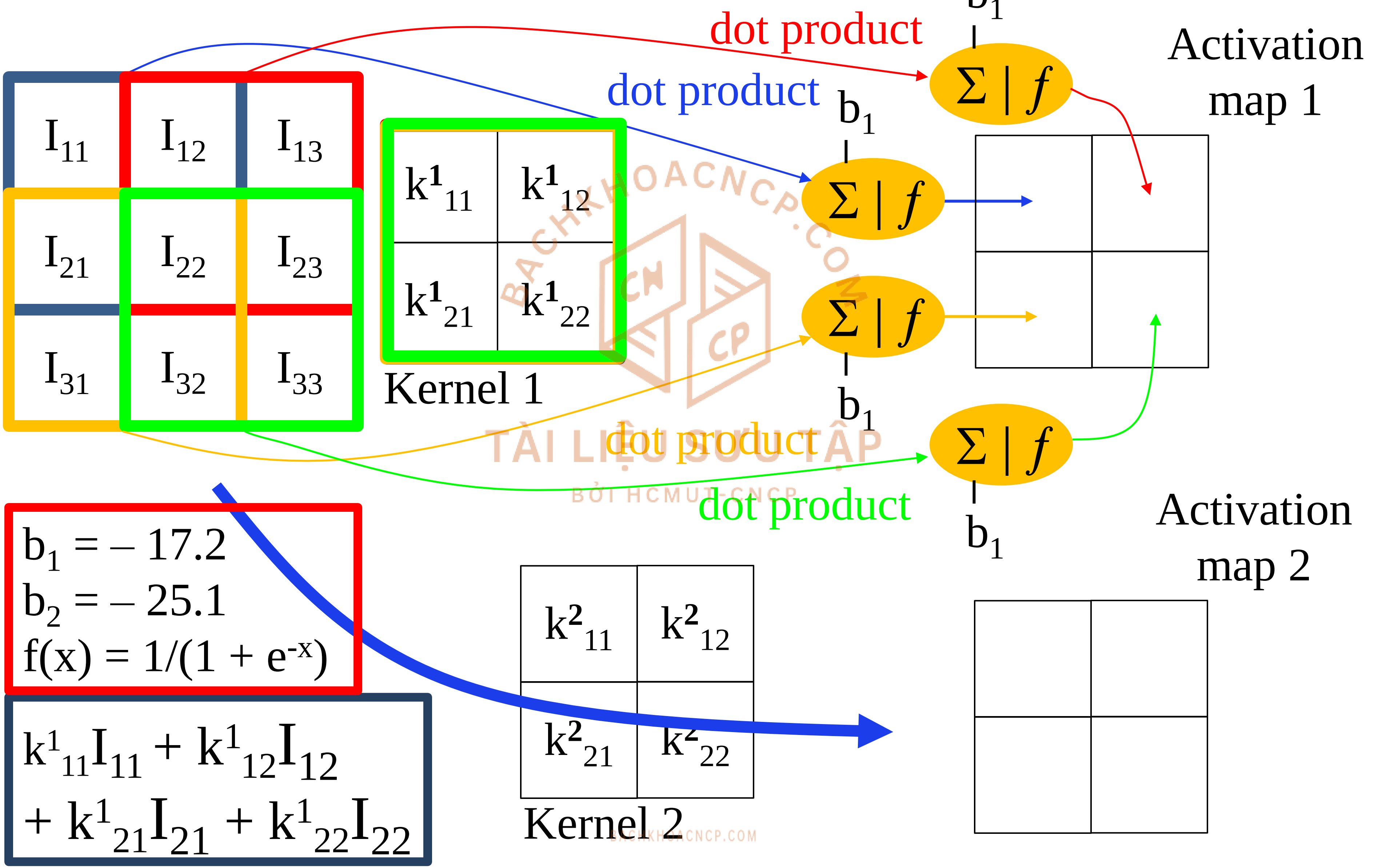
255	0	255
0	255	0
255	0	255

0	255	0
255	0	255
0	255	0

Convolution layer

255	0
0	255

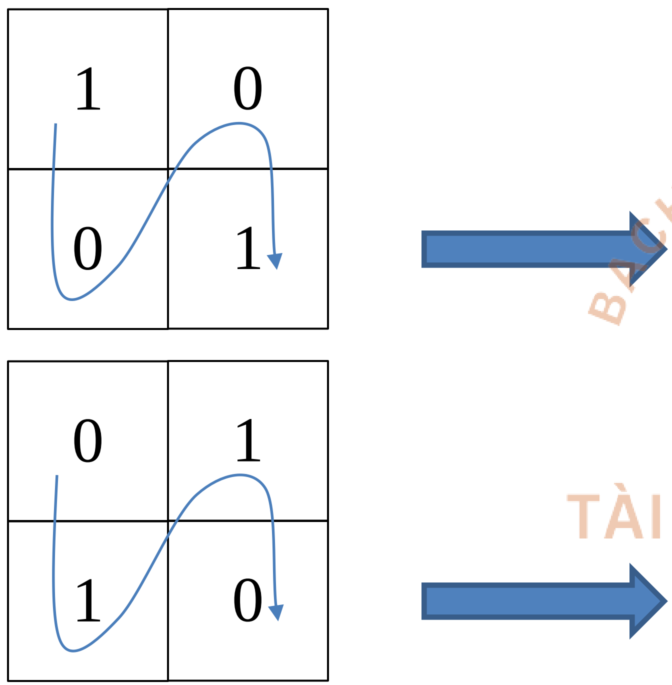
0	255
255	0

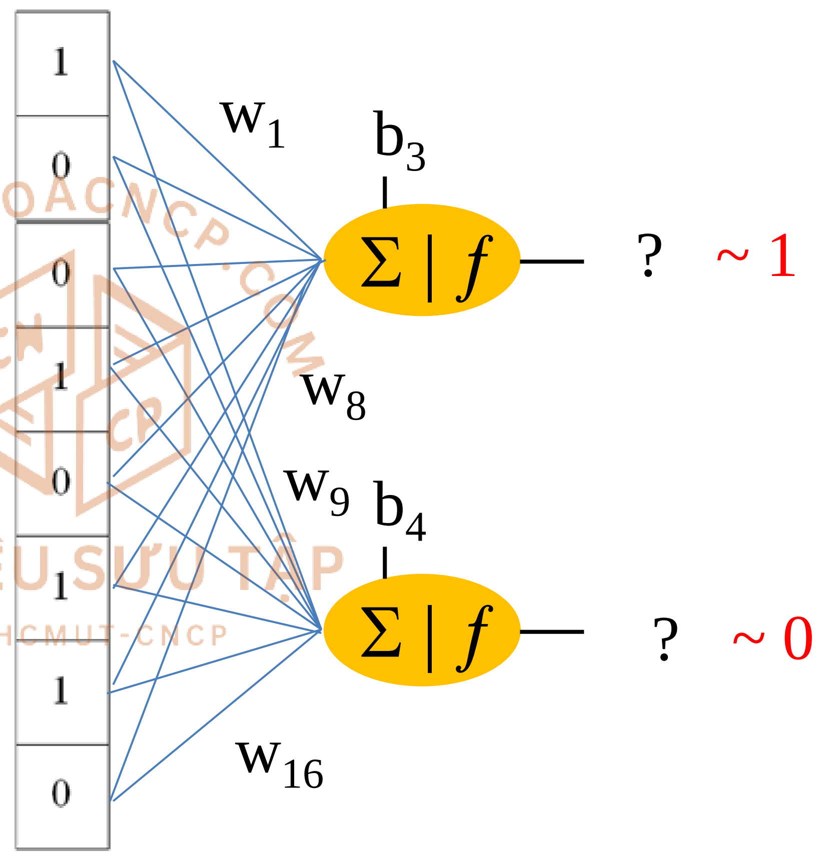


BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP

A toy example (forward)

Case 1: X





Activation map 1 & 2

$w_1, \ldots, w_8$= [76.7 -76.2 -91.4 24.8 -92.7 24.3 -14.0 -71.3]

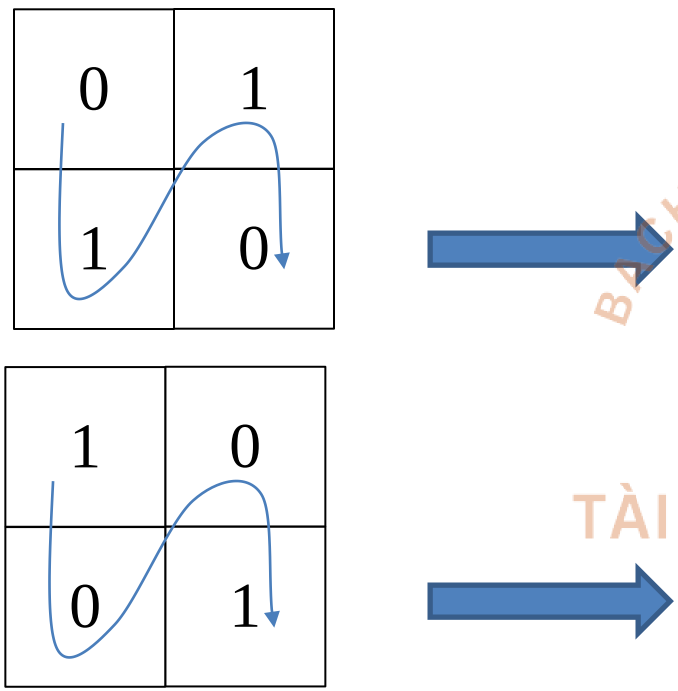
$w_9, \ldots, w_{16}$= [-98.9 69.6 77.5 -96.6 18.9 -53.8 -96.0 99.4]

b_3, b_4= [-47.9 -60.0]

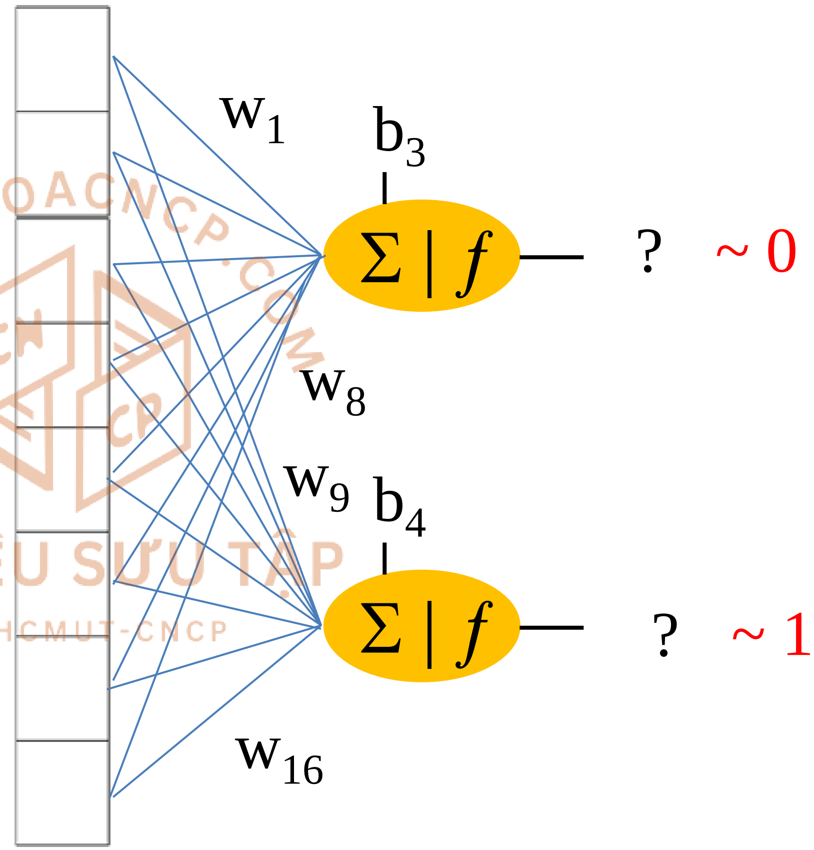
BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

A toy example (forward)

Case 2: O



Activation map 1 & 2



$w_1, \ldots, w_8$= [76.7 -76.2 -91.4 24.8 -92.7 24.3 -14.0 -71.3]

$w_8, \ldots, w_{16}$= [-98.9 69.6 77.5 -96.6 18.9 -53.8 -96.0 99.4]

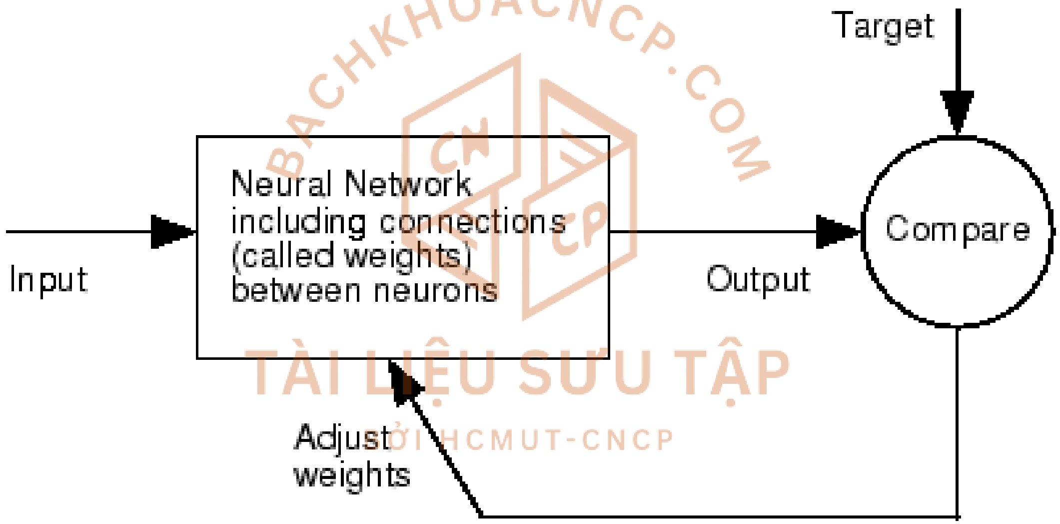
$b_3, \ldots, b_4$= [-47.9 -60.0]

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Questions

Q: Where do all parameters come from?

A: Backpropagation

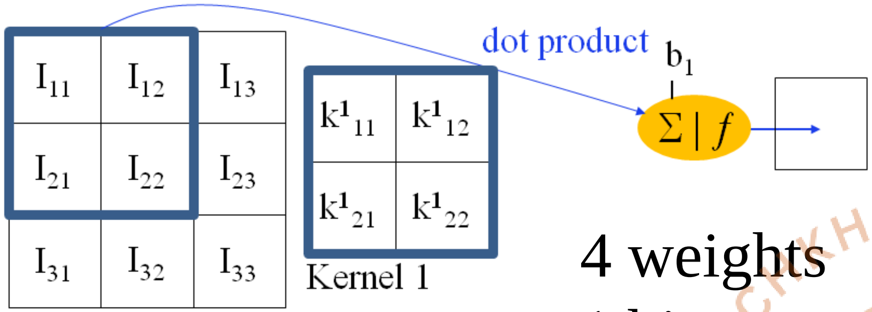


Q: Why local connectivity?

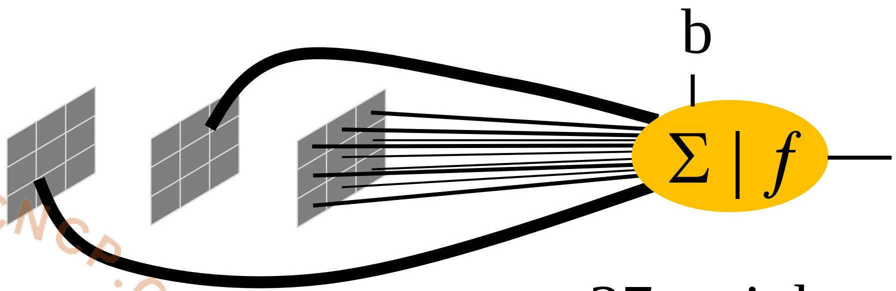
Q: Why shared weights?

BACHKHOACNCP.COM

General cases

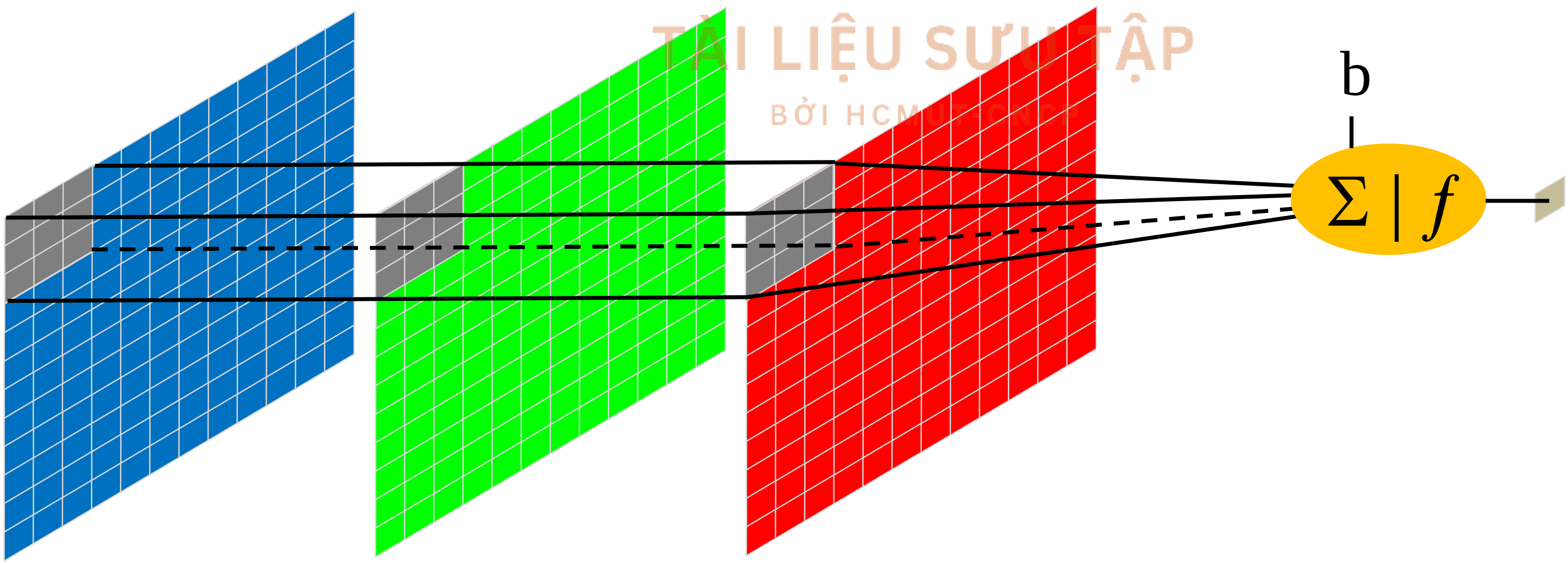


4 weights
1 bias



27 weights
1 bias

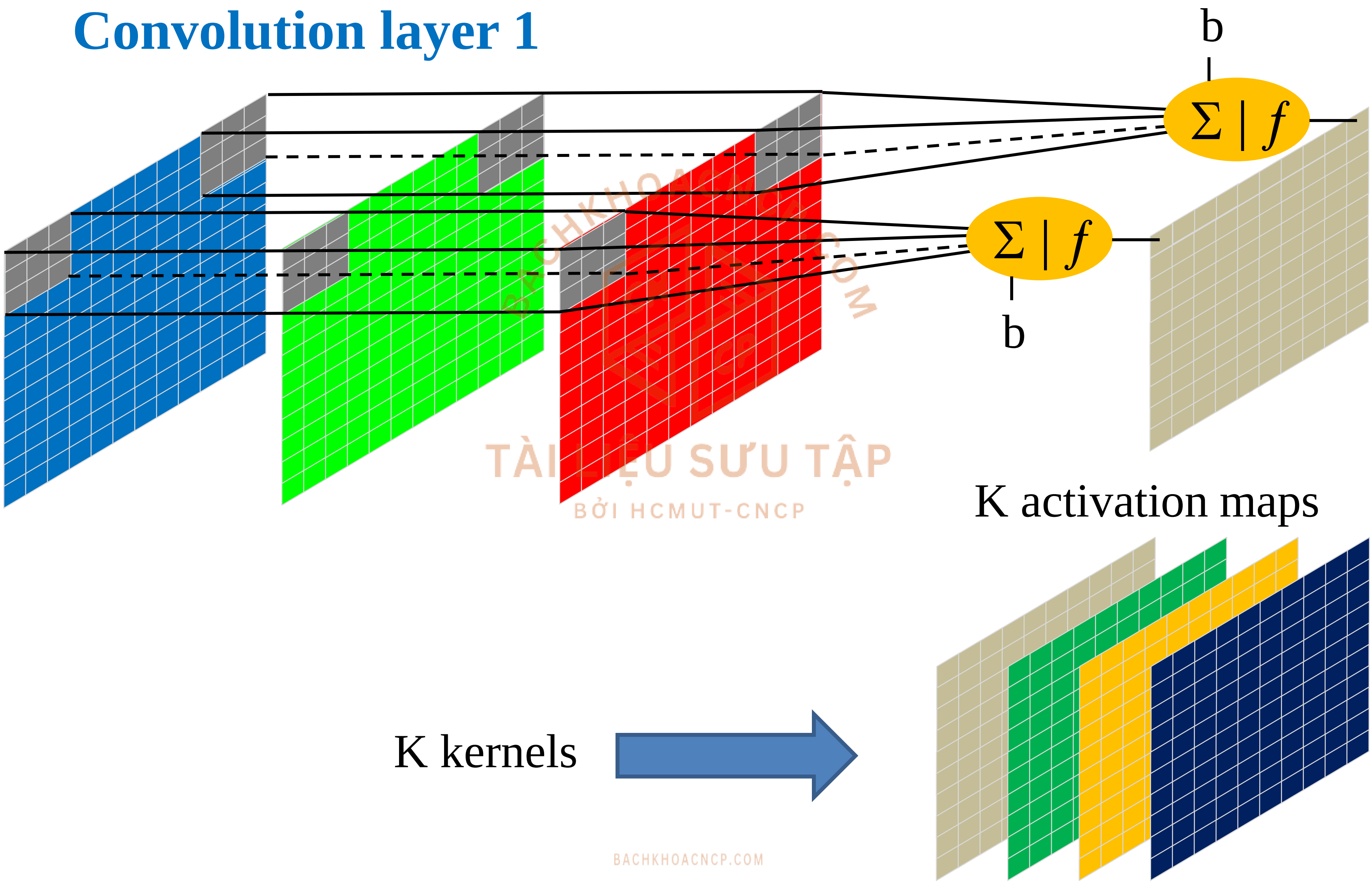
R, G, B layers



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

General cases

Convolution layer 1



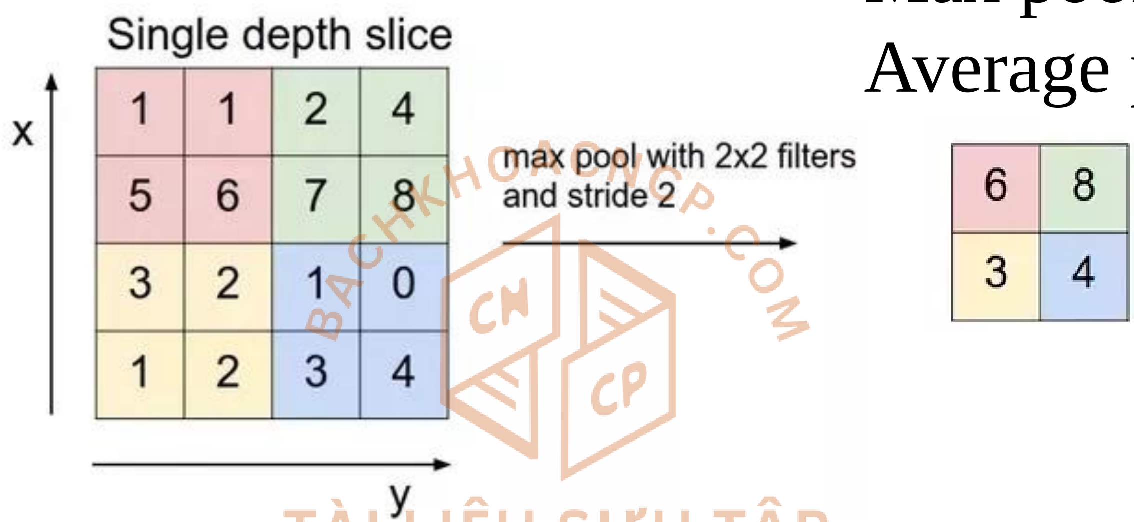
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

General cases

Pooling/Subsample layer

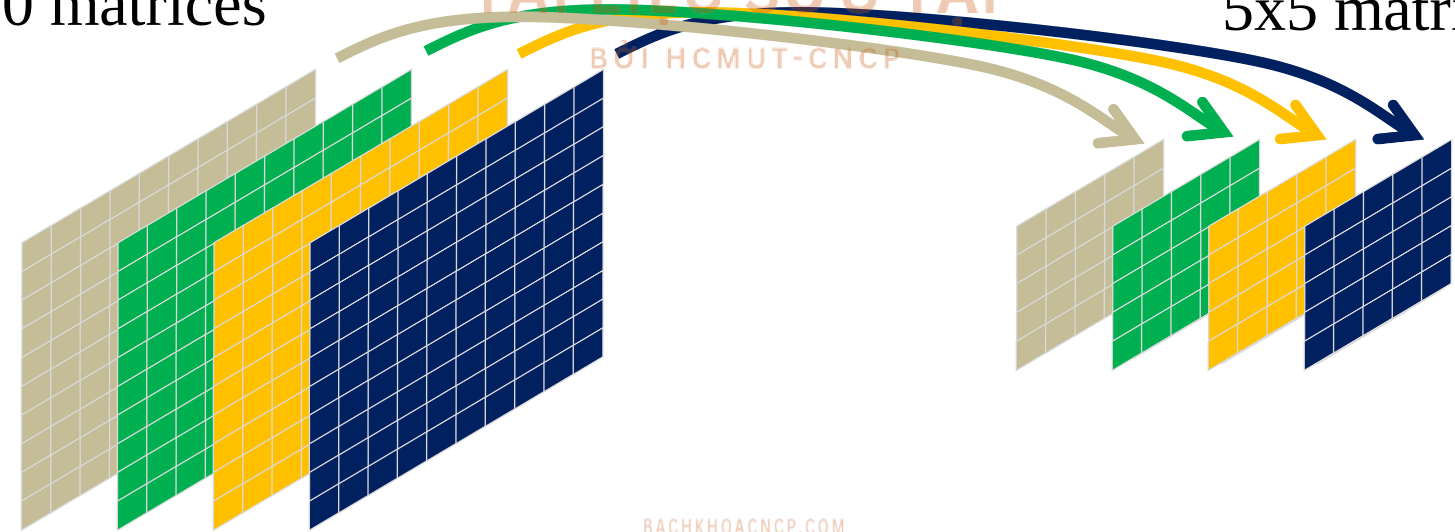
Max pooling

Average pooling



10x10 matrices

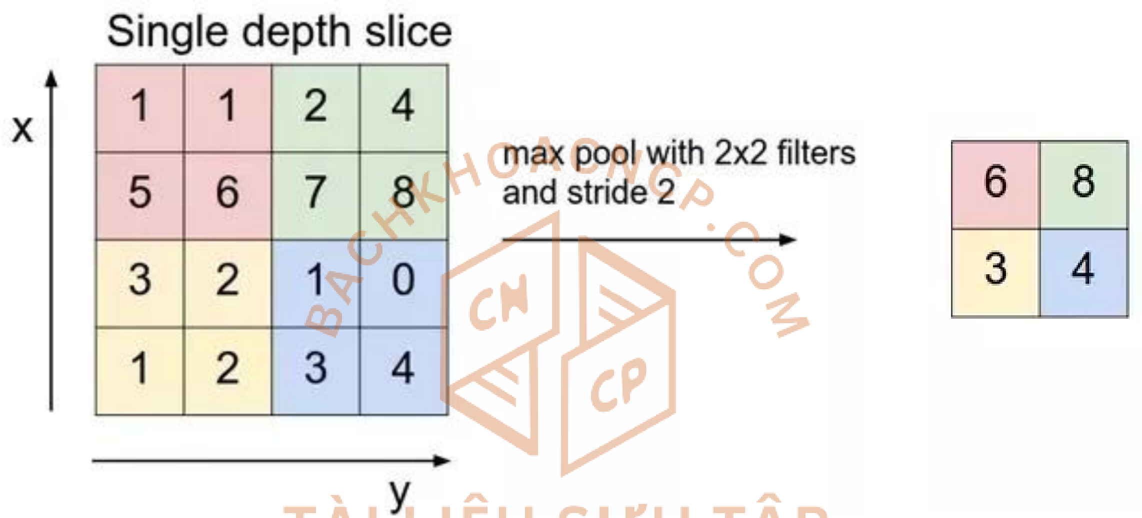
5x5 matrices



TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

General cases

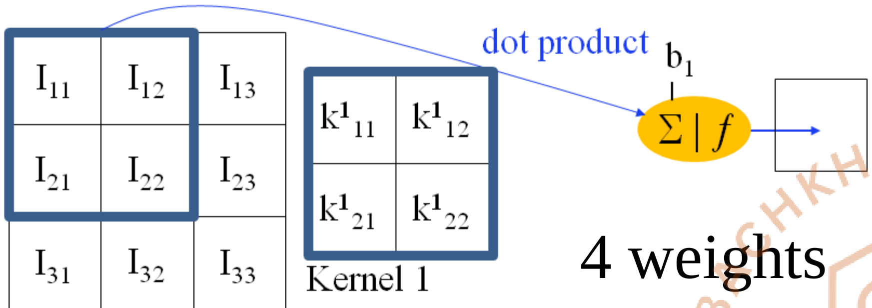
Pooling/Subsample layer



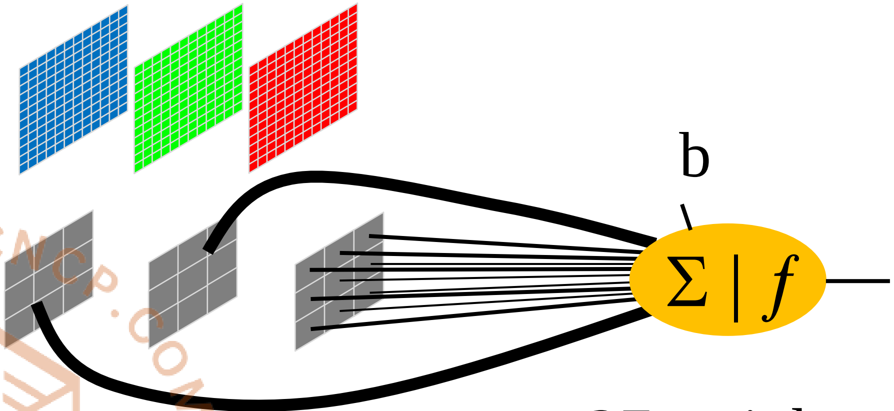
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

General cases

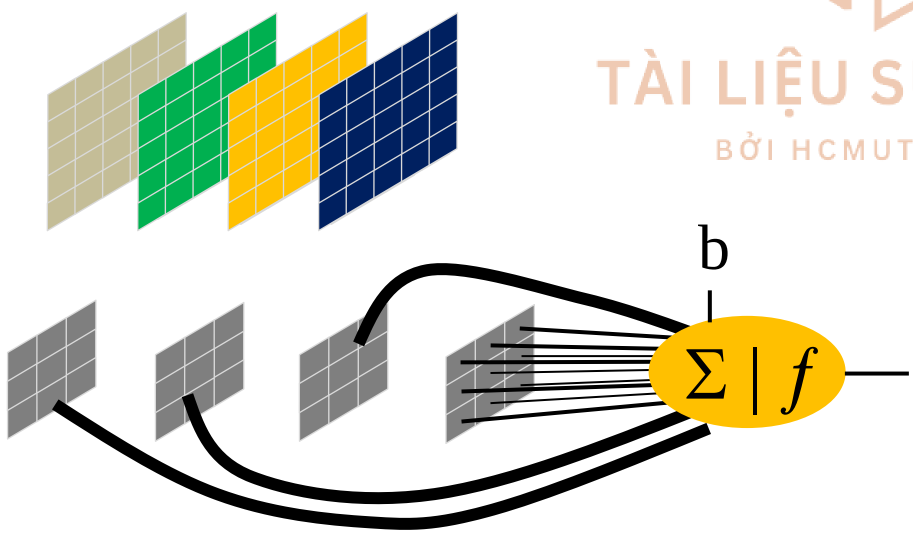
Convolutional layer 2



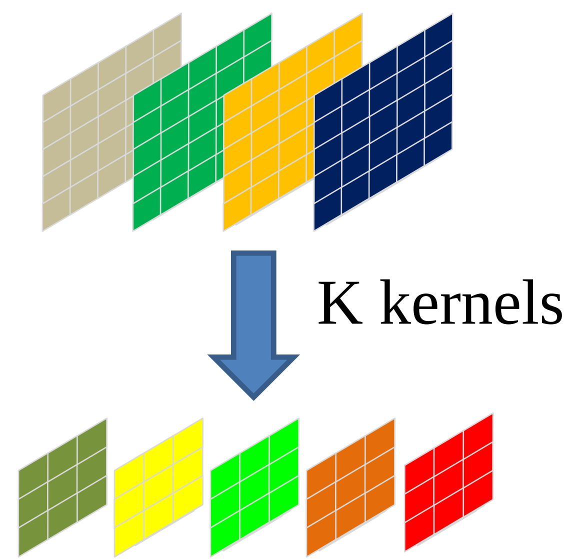
4 weights
1 bias



27 weights
1 bias



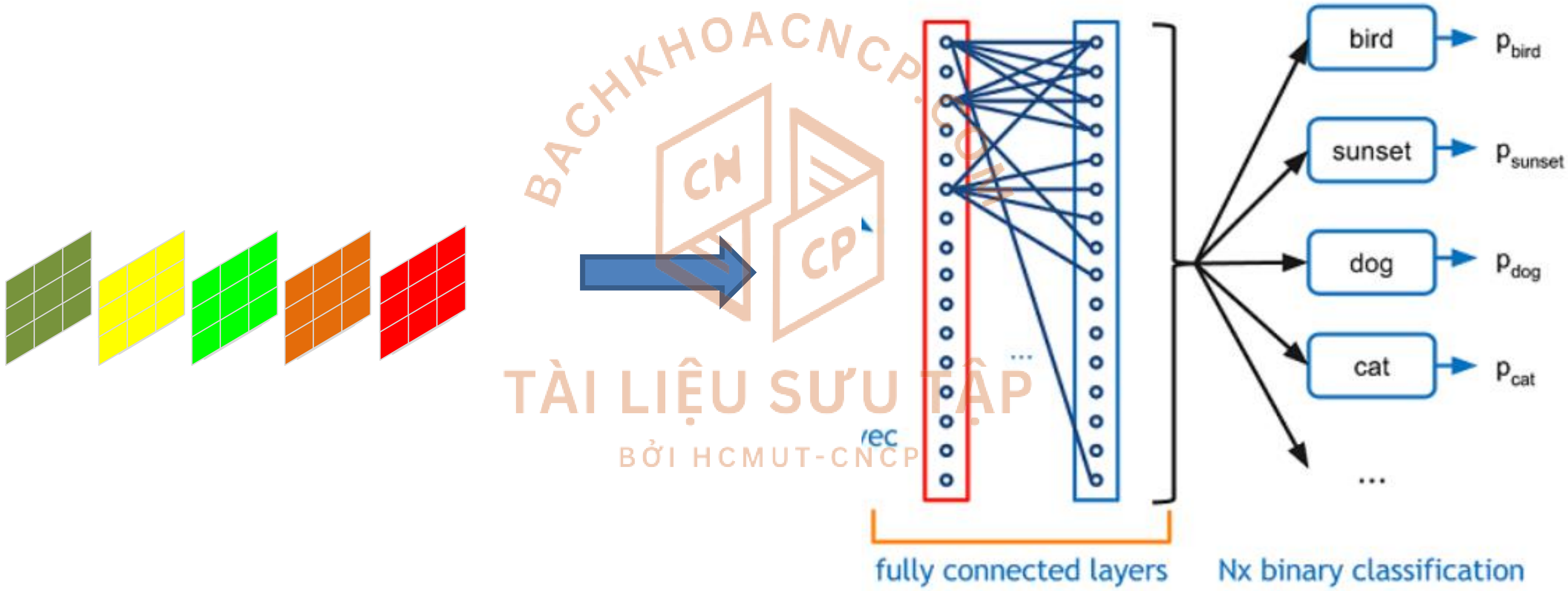
36 weights 1 bias



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

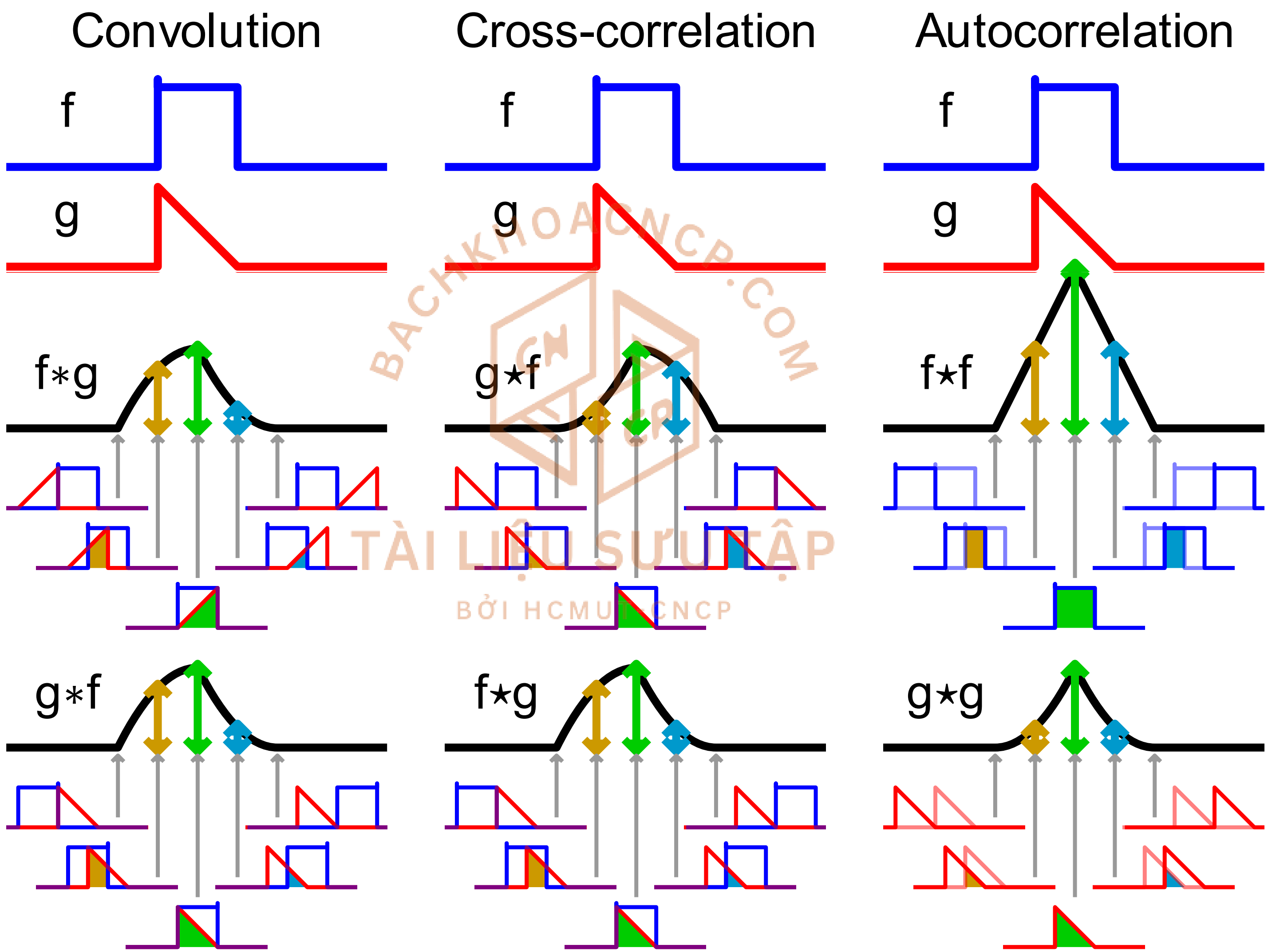
General cases

Fully Connected Layer



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

Convolution

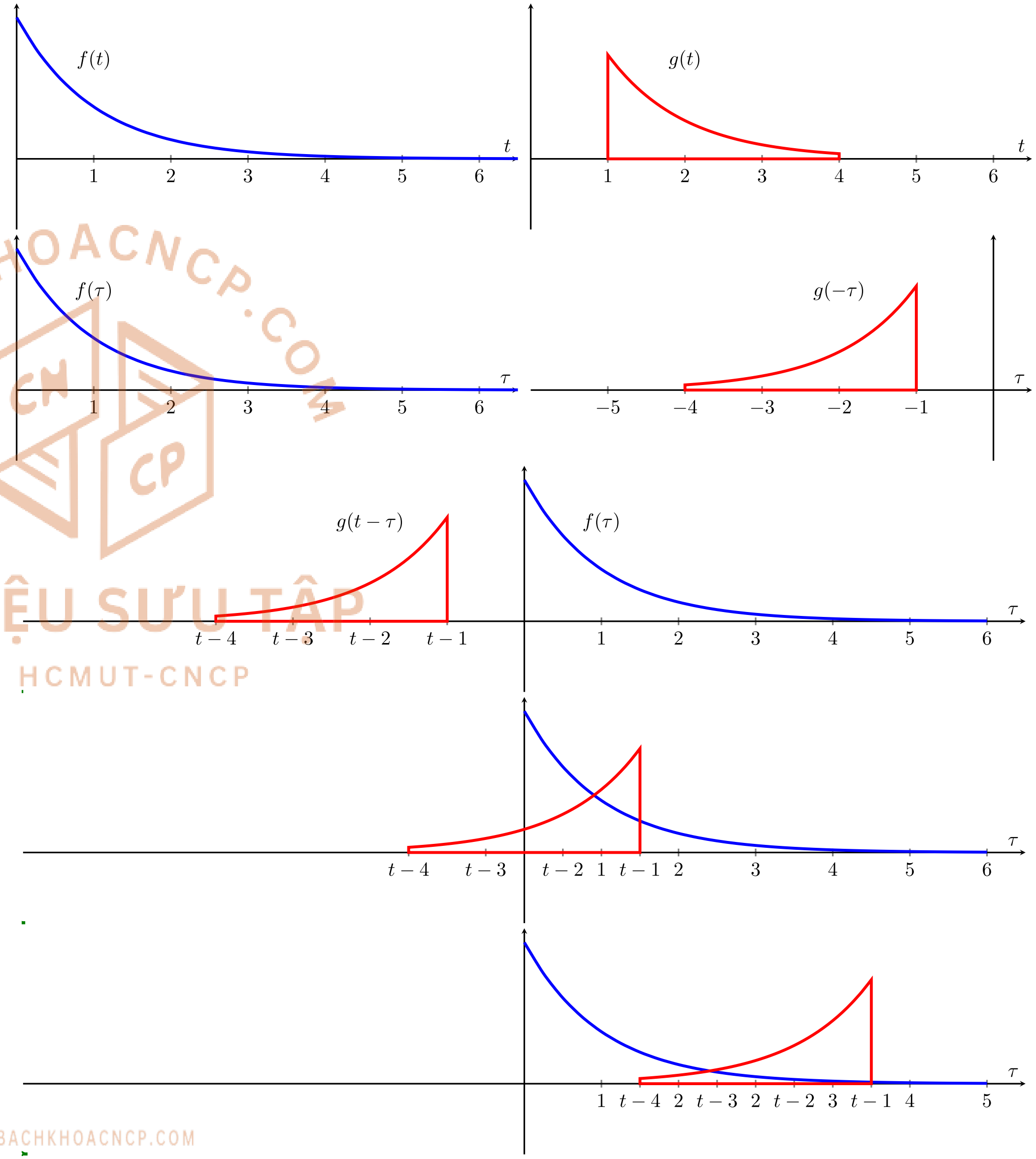


By Cmglee - Own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=20206883

Convolution

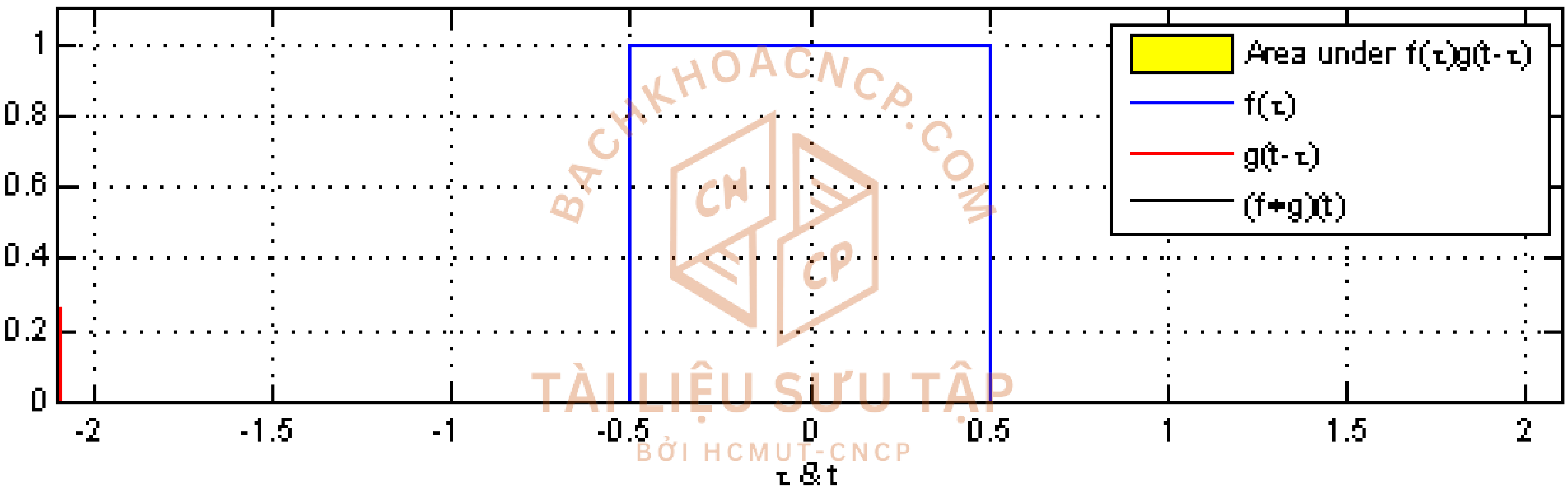
$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau$$

1. Express each function in terms of a dummy variable τ.
2. Reflect one of the functions: $g(\tau) \to g(-\tau)$.
3. Add a time-offset, t, which allows $g(t-\tau)$ to slide along the τ-axis.
4. Start t at $-\infty$ and slide it all the way to $+\infty$. Wherever the two functions intersect, find the integral of their product. In other words, at time t, compute the area under the function $f(\tau)$ weighted by the weighting function $g(t-\tau)$.



BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM

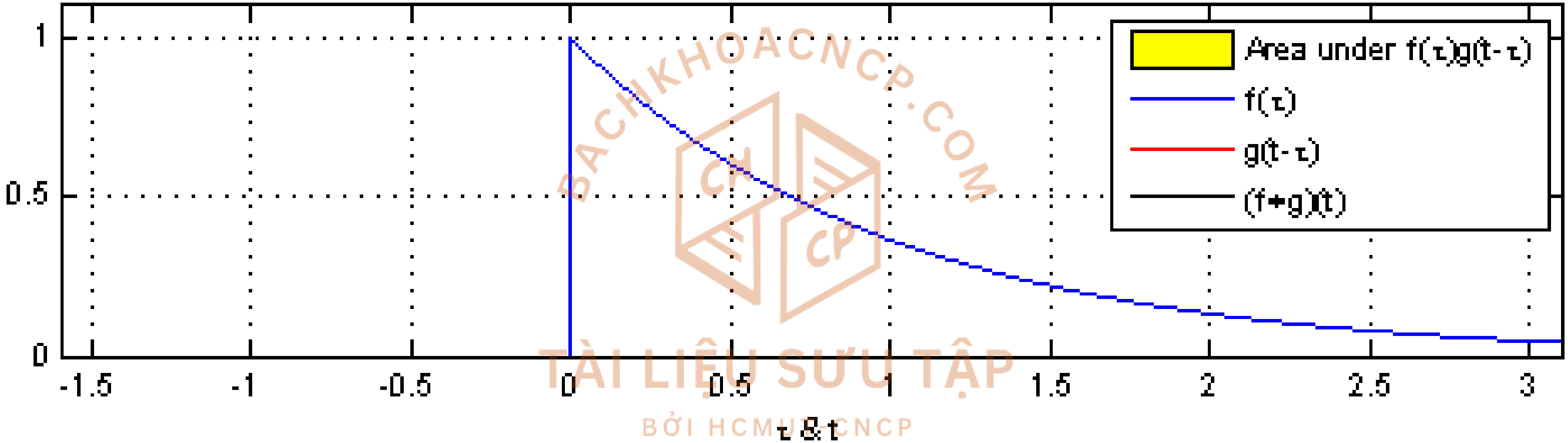
Convolution



By Convolution_of_box_signal_with_itself.gif: Brian Ambergderivative work: Tinos (talk) - Convolution_of_box_signal_with_itself.gif, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=11003835

BACHKHOACNCP.COM

Convolution



By Convolution_of_spiky_function_with_box.gif: Brian Ambergderivative work: Tinos (talk) - Convolution_of_spiky_function_with_box.gif, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=11003944

BACHKHOACNCP.COM

Convolution

BACHKHOACNCP.COM
TÀI LIỆU SƯU TẬP
BỞI HCMUT-CNCP
BACHKHOACNCP.COM